Boundary Conditions

To create a stellar model, one must solve four interrelated differential equations, which describe how the radius, pressure, luminosity, and temperature change with mass. To do this, we need four boundary conditions (to start off each differential). The boundary conditions for mass and luminosity are obvious:

$$r(m = 0) = 0 \tag{4.1.1}$$

$$\mathcal{L}(m = 0) = 0 \tag{4.1.2}$$

Unfortunately, we do not have any constraints for the central temperature and pressure. For these variables, we must apply boundary conditions at the surface of the star. To first order, we will define the stellar "surface" as the place where the optical depth, $\tau = 1$. At this location, $m = \mathcal{M}_T$, and we can apply the boundary conditions

$$P(m = \mathcal{M}_T) = 0 \tag{4.1.3}$$

$$T(m = \mathcal{M}_T) = 0 \tag{4.1.4}$$

This approximation isn't too bad, considering the temperatures and pressures in the interior of the star. A slightly better approximation is to use the equation

$$\mathcal{L}_T = 4\pi R^2 \sigma T_{\rm eff}^4 \tag{4.1.5}$$

(which actually defines effective temperature) and realize that the light we see must come from some finite optical depth

$$\tau = \int_R^\infty \kappa\, \rho\, dr = \bar{\kappa} \int_R^\infty \rho\, dr$$

and note that

$$P(R) = \int_R^\infty g\rho dr = g_0 \int_R^\infty \rho dr$$

Setting these two equations equal yields

$$P(R) = g_0 \tau / \bar{\kappa} = \frac{G\mathcal{M}_T}{R^2} \frac{\tau}{\bar{\kappa}} \tag{4.1.6}$$

where, again, $\tau \sim 1$. (Actually, it's 2/3.)

An even better approximation can be obtained using relations between surface temperature, pressure, luminosity, and radius that come from the calculations of stellar atmospheres. Grids of model stellar atmospheres exist that give values for P and T at their base, as a function for $\log g$ and $\mathcal{L}$. By interpolating in these grids, boundary conditions for P and T can be provided.

Expansions for the Center

A quick inspection of the equations of stellar structure demonstrates that they cannot be evaluated at $\mathcal{M} = 0$, due to powers of r and $\mathcal{M}$ in the denominator. For these locations, you need to use an expansion of each equation. For the continuity equation (2.1.3), this means

$$dr = \frac{1}{4\pi r^2 \rho} d\mathcal{M} \quad \Longrightarrow \quad r^2 dr = \frac{1}{4\pi\rho} d\mathcal{M}$$

If we assume that very close to the center, $\rho \approx \rho^0$, and we apply the boundary condition $r = 0$ at $\mathcal{M} = 0$, then

$$\int_0^r r^2 dr = \int_0^{\mathcal{M}} \frac{1}{4\pi\rho} d\mathcal{M} \quad \Longrightarrow \quad \frac{r^3}{3} = \frac{\mathcal{M}}{4\pi\rho^0}$$

or

$$r = \left(\frac{3\mathcal{M}}{4\pi\rho^0} \right)^{1/3} \tag{4.3.1}$$

Similarly, if the gradient in the energy sources is small, (2.3.3) expands to

$$d\mathcal{L} = (\epsilon_n - \epsilon_\nu + \epsilon_g)\, d\mathcal{M} \quad \Longrightarrow \quad \mathcal{L} = (\epsilon_n^0 - \epsilon_\nu^0 + \epsilon_g^0)\, \mathcal{M} \tag{4.3.2}$$

with its boundary condition.

For the momentum equation (2.2.9), we can assume that the center of the star is in mechanical equilibrium, so

$$dP = -\frac{G\mathcal{M}}{4\pi r^4} d\mathcal{M} = -\frac{G\mathcal{M}}{4\pi} \left(\frac{4\pi\rho^0}{3\mathcal{M}} \right)^{4/3} d\mathcal{M}$$

where we have substituted for radius using (4.3.1). Integrating this yields

$$P - P^0 = -\frac{G}{4\pi}\left(\frac{4\pi\rho^0}{3}\right)^{\frac{4}{3}} \int_0^{\mathcal{M}} \mathcal{M}^{-\frac{1}{3}} d\mathcal{M}$$

or

$$P - P^0 = -\frac{3G}{8\pi}\left(\frac{4\pi\rho^0}{3}\right)^{4/3} M^{2/3} \tag{4.3.3}$$

In the case of radiative energy transport, we can expand the temperature equation (3.1.5) to

$$dT = -\frac{3\kappa\mathcal{L}d\mathcal{M}}{64\pi^2 acr^4T^3} = -\frac{3\kappa^0(\epsilon_n^0 - \epsilon_\nu^0 + \epsilon_g^0)\mathcal{M}}{64\pi^2 acT^3}\left(\frac{4\pi\rho^0}{3\mathcal{M}}\right)^{4/3} d\mathcal{M}$$

where we have again substituted using (4.3.1) and (4.3.2). Keeping everything but the mass constant, we then get

$$\int_{T^0}^{T} T^3 dT = -\frac{\kappa^0(\epsilon_n^0 - \epsilon_\nu^0 + \epsilon_g^0)}{16\pi ac}\left(\frac{4\pi}{3}\right)^{\frac{1}{3}} \left\{\rho^0\right\}^{\frac{4}{3}} \int_0^{\mathcal{M}} M^{-\frac{1}{3}} d\mathcal{M}$$

yielding

$$T^4 - \{T^0\}^4 = -\frac{\kappa^0(\epsilon_n^0 - \epsilon_\nu^0 + \epsilon_g^0)}{2ac}\left(\frac{3}{4\pi}\right)^{2/3} \{\rho^0\}^{4/3}\mathcal{M}^{2/3} \tag{4.3.4}$$

Finally, for convective energy transport, we re-write (2.4.4) as

$$dT = -\frac{T}{P}\frac{G\mathcal{M}}{4\pi r^4}\nabla_{\rm ad}d\mathcal{M} = -\frac{T}{P^0}\frac{G\mathcal{M}}{4\pi}\left(\frac{4\pi\rho^0}{3\mathcal{M}}\right)^{4/3}\nabla^0_{\rm ad}$$

which implies

$$\int_{T^0}^{T}\frac{dT}{T} = -\frac{G\,\nabla^0_{\rm ad}(4\pi)^{1/3}}{3^{4/3}P^0}\{\rho^0\}^{4/3}\int_0^{\mathcal{M}}\mathcal{M}^{-1/3}d\mathcal{M}$$

and

$$\ln T - \ln T^0 = -\left(\frac{\pi}{6}\right)^{1/3}\frac{G\,\nabla^0_{\rm ad}}{P^0}\{\rho^0\}^{4/3}\mathcal{M}^{2/3} \qquad (4.3.5)$$

Solving the Equations

The equations of stellar structure are not analytic; they must be solved numerically. Furthermore, because the boundary conditions for the equations aren't all at the center, the differentials can't just be integrated outward.

The key to computing a stellar model is to divide a star up into K mass shells, with $10 < K < 1000$. The distribution of these shells is somewhat of an art form: in energy generating zones, where the temperature and density must be known with high accuracy, and near the surface, where the gradients are steep, the mass fraction must be small, $\sim 10^{-4}\mathcal{M}_T$. Zones as large as $0.1\mathcal{M}_T$ may be used elsewhere. In general, static stars need fewer shells than rapidly evolving stars.

THE MATCHING POINT METHOD

The original method of creating a numerical stellar model is the Matching Point, or Fitting method. In this procedure, one starts with the two center boundary conditions ($r = \mathcal{L} = 0$), but guesses the central temperature and pressure. The star is then integrated outward. At the same time an inward integration is performed, starting with the known surface boundary conditions (P_s and T_s) and guesses about R and $\mathcal{L}_T$. One then attempts to match the integrations somewhere in the middle of the star.

In general, of course, the two integrations won't match up: the guesses for the surface radius and luminosity and the central temperature and pressure will be wrong. At the matching point, the equations for the inward and outward integrations will differ by an amount Y. However, suppose the initial model is not too far off.

Let x_i be the initial guesses for R, $\mathcal{L}_T$, T_c, and P_c. (The index i, of course, goes from 1 to 4.) Perturb one of the initial guesses, such that $x_i' = x_i + \delta x_i$. Although the model still won't agree at the matching point, you can compare the old values of Y_i with the new values Y_i'. In other words, you obtain an estimate for $\frac{\partial Y_i}{\partial x_i}$ for each equation. Now compute similar partial derivatives for the other variables. You can now estimate the perturbations you need to match the inward and outward integrations, *i.e.,* to make $Y_i = 0$. This is done via 4 independent linear equations of the form

$$\left(\frac{\partial Y_1}{\partial x_1}\right)\delta x_1 + \left(\frac{\partial Y_1}{\partial x_2}\right)\delta x_2 + \left(\frac{\partial Y_1}{\partial x_3}\right)\delta x_3 + \left(\frac{\partial Y_1}{\partial x_4}\right)\delta x_4 = -Y_1$$

By solving the equations for the perturbations, δx_i, you can estimate the magnitude of the true perturbations needed to match things up. After a few iterations, the model should converge.

NUMERICAL TRICKS

Because of the large range of numbers involved, and the possibility of numerical roundoff in computers, there are a number of useful tricks that are sometimes used (and/or needed) in model calculations.

• Instead of using mass as the independent variable, sometimes one defines a new variable ξ, such that $\mathcal{M} = \mathcal{M}(\xi)$ and $0 < \xi < 1$. This changes the equations of stellar structure slightly, *i.e.,* mass conservations becomes

$$\frac{dr}{d\xi} = \frac{1}{4\pi r^2 \rho} \frac{d\mathcal{M}}{d\xi}$$

but makes the program more flexible, in that by changing $\mathcal{M}(\xi)$, you can immediately adjust to different mass distributions within a star. (For example, the distribution of mass within a main sequence star is completely different from that of a giant star.) By changing to ξ, the calculations can be made more linear.

• In computing derivatives by finite differences, some round off error may occur, especially if the variables are changing by many orders of magnitude. To minimize these errors, variables are generally made as linear as possible. For pressure and density, this means changing variables again, *i.e.,* substituting

$$p = P^{1/4} \qquad q = \rho^{1/2} \qquad F = L/\xi^2$$

Again, these substitutions change the character, but not the substance of the equations.

• In computing numerical derivatives, the most straightforward method is to just take the arithmetic average of quantities computed on each side of a shell. However, if the quantity being measured is increasing exponentially, an arithmetic average may not be appropriate: a geometrical average may be better. Unfortunately, there

is no one prescription for this problem, other than to use variables that are as linear as possible and to keep the shells thin.

• When computing the perturbations δr, δP, $\delta \mathcal{L}$, and δT, it is often times useful to "sneak up" on a solution, and limit your maximum correction to some small value. For example, if the equations predict that δr should be increased by 5%, it may be better to correct the value by only $\sim 1\%$. This will ensure that the perturbations stay in the linear regime. Without such a limit, the non-linearities in the equations may cause you to overcorrect, and continually bounce back and forth over the true solution.

The Henyey Method

The trouble with using the Matching Point method is that each model must be computed from scratch. While this may work for calculating a single model (say, a single main sequence star), it is inefficient for computing a series of models, as is done when modeling stellar evolution. Thus, the method of choice for computing a series of stellar models is the Henyey method, after Henyey, Forbes, & Gould 1964, *Ap. J.,* **139,** 306. (Other names for it include the relaxation method, the difference method, the shell method, and the generalized Newton-Raphson method.) To use the method, one first must start with a guess as to the structure of the star. (Usually, this is done by first computing a main-sequence model via the Matching Point Method.) Hopefully, this guess is not too far off – if it is, then the model will not coverge.

To see how the Henyey method works, we can start by re-writing all the spatial differentials as finite differences. In other words, (2.1.3), (2.2.9), (2.3.3), and (2.4.4) are written in the form of

$$\frac{dy}{dx} - f(x, y, z) = 0$$

The equations of stellar structure then become

$$\frac{r^{j+1}-r^j}{\mathcal{M}^{j+1}-\mathcal{M}^j}-\frac{1}{4\pi r^2\rho}=A_{j,1} \tag{4.4.1}$$

$$\frac{P^{j+1}-P^j}{\mathcal{M}^{j+1}-\mathcal{M}^j}+\frac{G\mathcal{M}}{4\pi r^4}+\left(\frac{1}{4\pi r^2}\right)\left(\frac{d^2r}{dt^2}\right)=A_{j,2} \tag{4.4.2}$$

$$\frac{\mathcal{L}^{j+1}-\mathcal{L}^j}{\mathcal{M}^{j+1}-\mathcal{M}^j}-\epsilon_n+\epsilon_\nu+c_P\frac{dT}{dt}-\frac{\delta}{\rho}\frac{dP}{dt}=A_{j,3} \tag{4.4.3}$$

$$\frac{T^{j+1}-T^j}{\mathcal{M}^{j+1}-\mathcal{M}^j}+\frac{T}{P}\frac{G\mathcal{M}}{4\pi r^4}\nabla=A_{j,4} \tag{4.4.4}$$

where the superscript j refers to the number of the shell, and $A_{i,j}=0$. Note that all the variables to the left of the equal signs are functions of $\mathcal{M}$, t, r, P, $\mathcal{L}$, and T, and they are (usually) evaluated half-way between shell j and $j+1$. Thus (4.4.1) is

$$\frac{r^{j+1}-r^j}{\mathcal{M}^{j+1}-\mathcal{M}^j}-\frac{1}{2}\left\{\frac{1}{4\pi\{r^{j+1}\}^2\rho^{j+1}}+\frac{1}{4\pi\{r^j\}^2\rho^j}\right\}=A_{j,1} \tag{4.4.5}$$

Note also that the variables ρ, c_P, δ, and $\nabla_{\rm ad}$ are known through the equation of state, ϵ_n, ϵ_ν, and κ are functions of ρ, T, and atomic physics, and the time derivatives can be written in terms of known values from the previous model, *i.e.*,

$$\frac{dP^j}{dt}=\frac{P^j-P_0^j}{\Delta t} \tag{4.4.6}$$

$$\frac{dT^j}{dt}=\frac{T^j-T_0^j}{\Delta t} \tag{4.4.7}$$

(If the second derivative term is needed, it is written via the velocity, which adds another variable and equation to the mix. The time steps for mechanical calculations, however, must be short, so that no shell has $\Delta r/\Delta t$ exceeding the sound speed.)

Next, we have the four special equations for the center of the star.

$$r^1 - \left(\frac{3\mathcal{M}^1}{4\pi\rho^0}\right)^{1/3} = A_{1,1} \tag{4.4.8}$$

$$P^1 - P^0 + \frac{3G}{8\pi}\left(\frac{4\pi\rho^0}{3}\right)^{4/3}\{M^1\}^{2/3} = A_{1,2} \tag{4.4.9}$$

$$\mathcal{L}^1 - (\epsilon_n^0 - \epsilon_\nu^0 + \epsilon_g^0)\,\mathcal{M}^1 = A_{1,4} \tag{4.4.10}$$

$$\{T^1\}^4 - \{T^0\}^4 + \frac{\kappa^0(\epsilon_n^0 - \epsilon_\nu^0 + \epsilon_g^0)}{2ac}\left(\frac{3}{4\pi}\right)^{\frac{2}{3}}\{\rho^0\}^{\frac{4}{3}}\{\mathcal{M}^1\}^{\frac{2}{3}} = A_{1,3} \tag{4.4.11}$$

or

$$\ln T^1 - \ln T^0 + \left(\frac{\pi}{6}\right)^{1/3} + \frac{G\,\nabla_{\rm ad}^0}{P^0}\{\rho^0\}^{4/3}\{\mathcal{M}^1\}^{2/3} = A_{1,4} \tag{4.4.12}$$

where again, the values $A_{i,j} = 0$. Note that in deriving (4.4.8) and (4.4.9) we have applied two boundary conditions, $r(0) = \mathcal{L}(0) = 0$.

Finally, we have the two outer boundary conditions:

$$P^K = B_P \tag{4.4.13}$$

$$T^K = B_T \tag{4.4.14}$$

where B_P and B_T are either zero, or obtained from $\mathcal{L}^K$ and r^K using a look-up table of stellar atmospheres. (Recall, however, that

normally the interior integration isn't carried out all the way to the surface; instead a match point is chosen interior to the region where partial ionization, super-adiabaticity, and changing opacities become important.)

Using the above equations, and the initial guess of the results, we can proceed with calculating a stellar model. Suppose our initial guess of the stellar model is wrong. In that case, the values $A_{i,j}$ in equations (4.4.1) through (4.4.4) and (4.4.8) through (4.4.13) will not be zero. We will therefore need to change our values of r, P, $\mathcal{L}$, and T by δr, δP, $\delta \mathcal{L}$, and δT. By how much? If we are not too far off, then linear theory should hold, and, for example

$$\left(\frac{\partial A_{j,1}}{\partial r^j}\right)\delta r^j + \left(\frac{\partial A_{j,1}}{\partial P^j}\right)\delta P^j + \left(\frac{\partial A_{j,1}}{\partial \mathcal{L}^j}\right)\delta \mathcal{L}^j +$$

$$\left(\frac{\partial A_{j,1}}{\partial T^j}\right)\delta T^j + \left(\frac{\partial A_{j,1}}{\partial r^{j+1}}\right)\delta r^{j+1} + \left(\frac{\partial A_{j,1}}{\partial P^{j+1}}\right)\delta P^{j+1} +$$

$$\left(\frac{\partial A_{j,1}}{\partial \mathcal{L}^{j+1}}\right)\delta \mathcal{L}^{j+1} + \left(\frac{\partial A_{j,1}}{\partial T^{j+1}}\right)\delta T^{j+1} = -A_{j,1} \qquad (4.4.15)$$

There are 4 equations like this for each mass shell. Except for shell 1, each equation contains 8 partial derivatives. (The innermost contains only six partials, since the values of r and $\mathcal{L}$ at the center are known; see (4.4.8) through (4.4.12)). In addition, we have the two equations (with four partials each) that come from the outer boundary condition. Now note: each partial derivative can be computed directly using the values of the initial guess, and the values of $A_{i,j}$ are already known. Thus, the only unknowns in (4.4.15) are the values of δr, δP, $\delta \mathcal{L}$, and δT.

We can write the series of (4.4.15)-type equations in matrix form. Suppose we have a (minimal) three shell model. The relations between the partial derivatives and $A_{i,j}$, B_P, and B_T is

$$\begin{pmatrix}
\frac{\partial A_{1,1}}{\partial P^0} & \frac{\partial A_{1,1}}{\partial T^0} & \frac{\partial A_{1,1}}{\partial r^1} & \frac{\partial A_{1,1}}{\partial P^1} & \frac{\partial A_{1,1}}{\partial \mathcal{L}^1} & \frac{\partial A_{1,1}}{\partial T^1} & & & & & & & & & & \\
\frac{\partial A_{1,2}}{\partial P^0} & \frac{\partial A_{1,2}}{\partial T^0} & \frac{\partial A_{1,2}}{\partial r^1} & \frac{\partial A_{1,2}}{\partial P^1} & \frac{\partial A_{1,2}}{\partial \mathcal{L}^1} & \frac{\partial A_{1,2}}{\partial T^1} & & & & & & & & & & \\
\frac{\partial A_{1,3}}{\partial P^0} & \frac{\partial A_{1,3}}{\partial T^0} & \frac{\partial A_{1,3}}{\partial r^1} & \frac{\partial A_{1,3}}{\partial P^1} & \frac{\partial A_{1,3}}{\partial \mathcal{L}^1} & \frac{\partial A_{1,3}}{\partial T^1} & & & & & & & & & & \\
\frac{\partial A_{1,4}}{\partial P^0} & \frac{\partial A_{1,4}}{\partial T^0} & \frac{\partial A_{1,4}}{\partial r^1} & \frac{\partial A_{1,4}}{\partial P^1} & \frac{\partial A_{1,4}}{\partial \mathcal{L}^1} & \frac{\partial A_{1,4}}{\partial T^1} & & & & & & & & & & \\
 & & \frac{\partial A_{2,1}}{\partial r^1} & \frac{\partial A_{2,1}}{\partial P^1} & \frac{\partial A_{2,1}}{\partial \mathcal{L}^1} & \frac{\partial A_{2,1}}{\partial T^1} & \frac{\partial A_{2,1}}{\partial r^2} & \frac{\partial A_{2,1}}{\partial P^2} & \frac{\partial A_{2,1}}{\partial \mathcal{L}^2} & \frac{\partial A_{2,1}}{\partial T^2} & & & & & & \\
 & & \frac{\partial A_{2,2}}{\partial r^1} & \frac{\partial A_{2,2}}{\partial P^1} & \frac{\partial A_{2,2}}{\partial \mathcal{L}^1} & \frac{\partial A_{2,2}}{\partial T^1} & \frac{\partial A_{2,2}}{\partial r^2} & \frac{\partial A_{2,2}}{\partial P^2} & \frac{\partial A_{2,2}}{\partial \mathcal{L}^2} & \frac{\partial A_{2,2}}{\partial T^2} & & & & & & \\
 & & \frac{\partial A_{2,3}}{\partial r^1} & \frac{\partial A_{2,3}}{\partial P^1} & \frac{\partial A_{2,3}}{\partial \mathcal{L}^1} & \frac{\partial A_{2,3}}{\partial T^1} & \frac{\partial A_{2,3}}{\partial r^2} & \frac{\partial A_{2,3}}{\partial P^2} & \frac{\partial A_{2,3}}{\partial \mathcal{L}^2} & \frac{\partial A_{2,3}}{\partial T^2} & & & & & & \\
 & & \frac{\partial A_{2,4}}{\partial r^1} & \frac{\partial A_{2,4}}{\partial P^1} & \frac{\partial A_{2,4}}{\partial \mathcal{L}^1} & \frac{\partial A_{2,4}}{\partial T^1} & \frac{\partial A_{2,4}}{\partial r^2} & \frac{\partial A_{2,4}}{\partial P^2} & \frac{\partial A_{2,4}}{\partial \mathcal{L}^2} & \frac{\partial A_{2,4}}{\partial T^2} & & & & & & \\
 & & & & & & \frac{\partial A_{3,1}}{\partial r^2} & \frac{\partial A_{3,1}}{\partial P^2} & \frac{\partial A_{3,1}}{\partial \mathcal{L}^2} & \frac{\partial A_{3,1}}{\partial T^2} & \frac{\partial A_{3,1}}{\partial r^3} & \frac{\partial A_{3,1}}{\partial P^3} & \frac{\partial A_{3,1}}{\partial \mathcal{L}^3} & \frac{\partial A_{3,1}}{\partial T^3} \\
 & & & & & & \frac{\partial A_{3,2}}{\partial r^2} & \frac{\partial A_{3,2}}{\partial P^2} & \frac{\partial A_{3,2}}{\partial \mathcal{L}^2} & \frac{\partial A_{3,2}}{\partial T^2} & \frac{\partial A_{3,2}}{\partial r^3} & \frac{\partial A_{3,2}}{\partial P^3} & \frac{\partial A_{3,2}}{\partial \mathcal{L}^3} & \frac{\partial A_{3,2}}{\partial T^3} \\
 & & & & & & \frac{\partial A_{3,3}}{\partial r^2} & \frac{\partial A_{3,3}}{\partial P^2} & \frac{\partial A_{3,3}}{\partial \mathcal{L}^2} & \frac{\partial A_{3,3}}{\partial T^2} & \frac{\partial A_{3,3}}{\partial r^3} & \frac{\partial A_{3,3}}{\partial P^3} & \frac{\partial A_{3,3}}{\partial \mathcal{L}^3} & \frac{\partial A_{3,3}}{\partial T^3} \\
 & & & & & & \frac{\partial A_{3,4}}{\partial r^2} & \frac{\partial A_{3,4}}{\partial P^2} & \frac{\partial A_{3,4}}{\partial \mathcal{L}^2} & \frac{\partial A_{3,4}}{\partial T^2} & \frac{\partial A_{3,4}}{\partial r^3} & \frac{\partial A_{3,4}}{\partial P^3} & \frac{\partial A_{3,4}}{\partial \mathcal{L}^3} & \frac{\partial A_{3,4}}{\partial T^3} \\
 & & & & & & & & & & \frac{\partial B_P}{\partial r^3} & \frac{\partial B_P}{\partial P^3} & \frac{\partial B_P}{\partial \mathcal{L}^3} & \frac{\partial B_P}{\partial T^3} \\
 & & & & & & & & & & \frac{\partial B_T}{\partial r^3} & \frac{\partial B_T}{\partial P^3} & \frac{\partial B_T}{\partial \mathcal{L}^3} & \frac{\partial B_T}{\partial T^3}
\end{pmatrix}
\begin{pmatrix} \delta P^0 \\ \delta T^0 \\ \delta r^1 \\ \delta P^1 \\ \delta \mathcal{L}^1 \\ \delta T^1 \\ \delta r^2 \\ \delta P^2 \\ \delta \mathcal{L}^2 \\ \delta T^2 \\ \delta r^3 \\ \delta P^3 \\ \delta \mathcal{L}^3 \\ \delta T^3 \end{pmatrix}
=
\begin{pmatrix} -A_{1,1} \\ -A_{1,2} \\ -A_{1,3} \\ -A_{1,4} \\ -A_{2,1} \\ -A_{2,2} \\ -A_{2,3} \\ -A_{2,4} \\ -A_{3,1} \\ -A_{3,2} \\ -A_{3,3} \\ -A_{3,4} \\ -B_P \\ -B_T \end{pmatrix}$$

or, in matrix notation

$$\mathbf{HU} = \mathbf{A} \tag{4.4.16}$$

where $\mathbf{H}$ is called the Henyey matrix. The unknown vector containing the correction values is then

$$\mathbf{U} = \mathbf{H}^{-1}\mathbf{A} \tag{4.4.17}$$

In fact, since the Henyey matrix is rather sparse (with many zeros, except along the diagonal), it's fairly easy to find efficient algorithms to compute its inverse. Once $\mathbf{U}$ is known, these correction values can be used to improve the original guess, and the procedure can be repeated. After several iterations, the values $A_{i,j}$, B_P, and B_T may approach zero, giving you the structure of the star.

Digression: Forward and Back Differencing Techniques

There are two ways to numerically solve a system of differential equations. The first method is through "forward differencing," or "explicit" numerical integration. With this technique, values for integrating step $n+1$ are estimated directly from the known values of the previous step. In the alternative "back differencing" or "implicit" integration method, the values at step $n+1$ are estimated using the $n+1$ values themselves, via numerical inversion.

For example, consider the simple differential equation

$$\frac{dX}{dt} = a - bX$$

which has the analytical solution

$$X = \frac{a}{b} + e^{-bt}$$

In the forward differencing method, this equation is

$$\frac{X_{n+1} - X_n}{\Delta t} = a - bX_n$$

which translates to

$$X_{n+1} = a\Delta t + X_n\left(1 - b\Delta t\right) \tag{4.2.1}$$

while in the back differencing method, the equation is

$$\frac{X_{n+1} - X_n}{\Delta t} = a - bX_{n+1}$$

or

$$X_{n+1} = \frac{a\Delta t + X_n}{1 + b\Delta t} \tag{4.2.2}$$

Now note that (4.2.1) and (4.2.2) behave very differently as the time step becomes large, *i.e.,* $\Delta t \to \infty$. With forward differencing, the numerical solution diverges; with back differencing, $X_{n+1} \to a/b$, which is the analytical result. This is characteristic of the two approaches. In general, forward differencing solutions are unstable, and can only be used with very small time steps (*i.e.,* for the above example, $b\Delta t < 1$). Back differencing methods are much more reliable, and should be used whenever possible.

Digression: Solving Initial Value Problems

A "standard" way to solve a set of differential equations with known initial values is through fourth-order Runge-Kutta integration. To appreciate the method, let us solve the simple differential equation

$$\frac{dy}{dx} = f(x,y) = x + y, \qquad y(0) = 1 \tag{A1.1}$$

using a Taylor expansion. If we expand $y(x)$ about $x_0 = 0$, the expression for the function becomes

$$y(x) = y(x_0) + y'(x_0)\Delta x + \frac{y''(x_0)}{2!}\Delta x^2 + \frac{y'''(x_0)}{3!}\Delta x^3 + \ldots \tag{A1.2}$$

where $\Delta x = (x - x_0)$. From the boundary condition, we know that $y(x_0) = 1$, hence from (A1.1),

$$y'(x_0) = f(x_0, y_0) = x_0 + y(x_0) = 0 + 1 = 1 \tag{A1.3}$$

Next, we take the derivative of both sides of (A1.1), to get

$$\frac{d^2y}{dx^2} = \frac{dx}{dx} + \frac{dy}{dx} \quad \Longrightarrow \quad y'' = 1 + y' \tag{A1.4}$$

From (A1.3) we know that $y'(x_0) = 1$, hence (A1.4) gives us

$$y''(x_0) = 1 + 1 = 2 \tag{A1.5}$$

Similarly, if we take the next derivative of the function,

$$y''' = 0 + y''$$

and evaluate it at $x = x_0$ using (A1.5), we get $y'''(x_0) = 2$. This process can be extended as far out as needed. In the present case, the evaluation of $y(x)$ yields

$$y(x) = 1 + \Delta x + (\Delta x)^2 + \frac{1}{3}(\Delta x)^3 + \mathcal{O}^4 \tag{A1.6}$$

where $\mathcal{O} \sim \Delta x$ is the error term.

Note here, that had we stopped the expansion after the first term (*i.e.,* keeping the relation linear), the expression for $y(x)$ would have been

$$y(x) = y(x_0) + y'(x_0)\Delta x = y(x_0) + f(x_0, y_0)\Delta x + \mathcal{O}^2 \qquad (A1.7)$$

This is just simple forward-differencing solution (or Euler's method). Obviously, from (A1.7), we can expect the errors associated with the technique to be $\sim (\Delta x)^2$ *per step.* If the entire interval is divided into $1/\Delta x$ steps, then we expect the total error in the numerical integration to be $\sim \Delta x$.

It is easy to see why Euler's method is not very accurate — the slope used to determine the new value of y always comes from the beginning of the interval, and thus is always wrong. A better estimate would come from using the slope of the function at the middle of the interval, *i.e.,*

$$y(x) = y(x_0) + \frac{y'(x_0) + y'(x)}{2}\Delta x \qquad (A1.8)$$

Of course this function cannot be evaluated, since we cannot evaluate $y'(x)$ without knowing $y(x)$. However, we can *estimate* $y(x)$ from Euler's method, and then use the predicted value of $y'(x)$ to produce a "corrected" value of $y(x)$. In other words

$$\begin{aligned} y'(x_0) &= f(x_0, y_0) \\ y_{\rm est}(x) &= y(x_0) + y'(x_0)\Delta x \\ y'_{\rm est}(x) &= f(x, y_{\rm est}) \\ y(x) &= y(x_0) + \frac{y'(x_0) + y'_{\rm est}(x)}{2}\Delta x \end{aligned} \qquad (A1.9)$$

(This procedure can be repeated again if necessary, although after a while, it's better just to cut down the step size.) The accuracy of this modified Euler technique is $\sim (\Delta x)^3$. To see this, simply compare (A1.8) with the Taylor series expansion (A1.2), while substituting a forward-differencing expression for $y''(x_0)$, *i.e.,*

$$\begin{aligned} y(x) &= y(x_0) + y'(x_0)\Delta x + \frac{1}{2}\left(\frac{y'(x) - y'(x_0)}{\Delta x}\right)\Delta x^2 + \mathcal{O}^3 \\ &= y(x_0) + y'(x_0)\Delta x + \frac{y'(x)}{2}\Delta x - \frac{y'(x_0)}{2}\Delta x + \mathcal{O}^3 \\ &= y(x_0) + \frac{y'(x) + y'(x_0)}{2}\Delta x + \mathcal{O}^3 \qquad (A1.10) \end{aligned}$$

As above this error is per step; the total error of the integration is Δx^2.

The next improvement comes by combining the results of several estimates along the interval and weighting each one differently. For instance, suppose we generalize (A1.8) to

$$\begin{aligned} &y(x) = y(x_0) + ak_1 + bk_2 \qquad\qquad \text{where} \\ &k_1 = y'(x_0, y_0)\Delta x \qquad k_2 = y'(x_0 + \alpha\Delta x, y_0 + \beta k_1)\Delta x \quad (A1.11) \end{aligned}$$

where a and b are the weights of the two estimates, α describes where on the interval the estimates are made, and β is a correction factor for the estimate made using k_1. When written out fully with the substitution $f(x, y) = y'$, (A1.11) becomes

$$y(x) = y(x_0) + af(x_0, y_0)\Delta x + bf(x_0 + \alpha\Delta x, y_0 + \beta f(x_0, y_0)\Delta x)\Delta x$$

If we expand the last term as a two-dimensional Taylor series about x_0, y_0, then

$$y(x) = y(x_0) + af(x_0, y_0)\Delta x + b\Big\{ f(x_0, y_0) + \frac{\partial f(x_0, y_0)}{\partial x}\alpha\Delta x + \frac{\partial f(x_0, y_0)}{\partial y}\beta f(x_0, y_0)\Delta x \Big\}\Delta x$$

$$= y(x_0) + (a+b)f(x_0, y_0)\Delta x + \alpha b\frac{\partial f(x_0, y_0)}{\partial x}\Delta x^2 + \beta b\frac{\partial f(x_0, y_0)}{\partial y}f(x_0, y_0)\Delta x^2 \qquad (A1.12)$$

This expression can then be compared to the Taylor expansion for $y(x)$ given in (A1.2)

$$y(x) = y(x_0) + f(x_0, y_0)\Delta x + \frac{1}{2}f'(x_0, y_0)\Delta x^2$$

$$= y(x_0) + f(x_0, y_0)\Delta x + \frac{\Delta x^2}{2}\left\{ \frac{\partial f(x_0, y_0)}{\partial x} + \frac{\partial f(x_0, y_0)}{\partial y}\frac{dy}{dx} \right\}$$

$$= y(x_0) + f(x_0, y_0)\Delta x + \frac{\Delta x^2}{2}\left\{ \frac{\partial f(x_0, y_0)}{\partial x} + \frac{\partial f(x_0, y_0)}{\partial y}f(x_0, y_0) \right\} \qquad (A1.13)$$

Thus, $a + b = 1$, $\alpha\, b = 1/2$, and $\beta\, b = 1/2$. These are the coefficients for the second-order Runge-Kutta method; note that since there are three equations and four unknowns, more than one set of coefficients will work.

The fourth-order Runge-Kutta method works the same way, except that four intermediate estimates are made. The equations for this method are

$$y(x) = y(x_0) + \frac{k_1}{6} + \frac{k_2}{3} + \frac{k_3}{3} + \frac{k_4}{6} \qquad \text{where}$$

$$k_1 = y'(x_0, y_0)\Delta x$$

$$k_2 = y'(x_0 + \frac{\Delta x}{2}, y_0 + \frac{k_1}{2})\Delta x$$

$$k_3 = y'(x_0 + \frac{\Delta x}{2}, y_0 + \frac{k_2}{2})\Delta x$$

$$k_4 = y'(x_0 + \Delta x, y_0 + k_3)\Delta x \qquad (A1.14)$$

In general, the method gives results with accuracies of the order of $(\Delta x)^5$ per step, and $(\Delta x)^4$ over the entire integration range. Because of its accuracy, and relatively few computational steps, it is perhaps the most widely used technique for solving initial-value differential equations.

A final improvement, given by Press in *Numerical Recipes,* is to use fourth-order Runge-Kutta with an adaptive step size. In this scheme, each integration is performed twice, with two different step sizes. For example, if you compute $y(x)$ first using a stepsize of Δx_1 and then using a stepsize of $\Delta x_1/2$, the difference in the results, ϵ_1, should be a measure of the local accuracy. Moreover, since the local error term on fourth-order Runge-Kutta integration goes as $(\Delta x)^5$,

$$\frac{\Delta x_0}{\Delta x_1} = \left(\frac{\epsilon_0}{\epsilon_1}\right)^{1/5}$$

where ϵ_0 is some *desired* accuracy, and Δx_0 is the stepsize that will yield that accuracy. Thus, to gain additional accuracy from a

Runge-Kutta method, you can compare your estimate of the local accuracy with some desired accuracy. If it is too large, you can try again with a smaller stepsize; if it is too small, you can speed things up by increasing Δx for the next step of the integration.